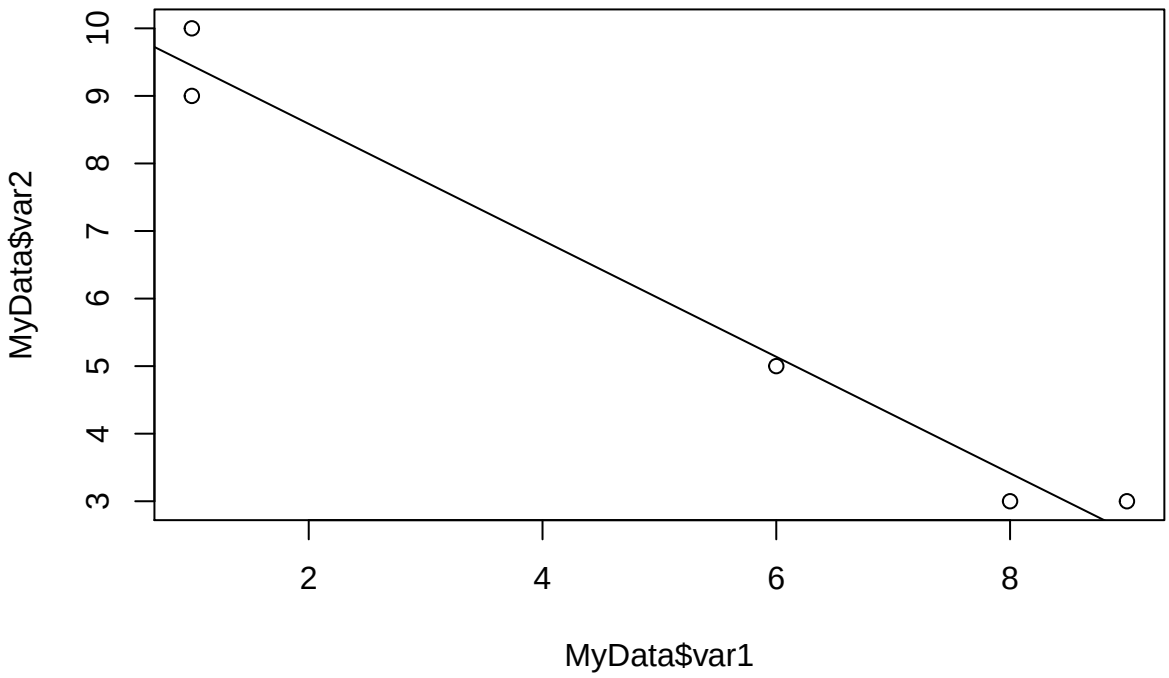
RegressionSimulator.R

Administrator

Tue Apr 05 15:01:29 2016

```
# Analyzing regression data
# 5 April 2016
# NJG


# eyeball data directly from a published scatter plot.
# each observation is a paired x,y data point
obs1 <- c(1,10)
obs2 <- c(9,3)
obs3 <- c(6,5)
obs4 <- c(1,9)
obs5 <- c(8,3)

MyData <- data.frame(rbind(obs1,obs2,obs3,obs4,obs5))
colnames(MyData) <- c("var1","var2")
plot(x=MyData$var1, y=MyData$var2)
MyModel <- lm(MyData$var2~MyData$var1)
z <- summary(MyModel)
abline(summary(MyModel))
```

```
## Warning in abline(summary(MyModel)): only using the first two of 8
## regression coefficients
```




```
# extract parameters from model object
print(z)
```

```
##
## Call:
## lm(formula = MyData$var2 ~ MyData$var1)
##
## Residuals:
##       1       2       3       4       5
##  0.5517  0.4483 -0.1379 -0.4483 -0.4138
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.31034    0.43426   23.74 0.000164 ***
## MyData$var1 -0.86207    0.07178  -12.01 0.001242 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5467 on 3 degrees of freedom
## Multiple R-squared:  0.9796, Adjusted R-squared:  0.9728
## F-statistic: 144.2 on 1 and 3 DF,  p-value: 0.001242
```

```
names(z)
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
```

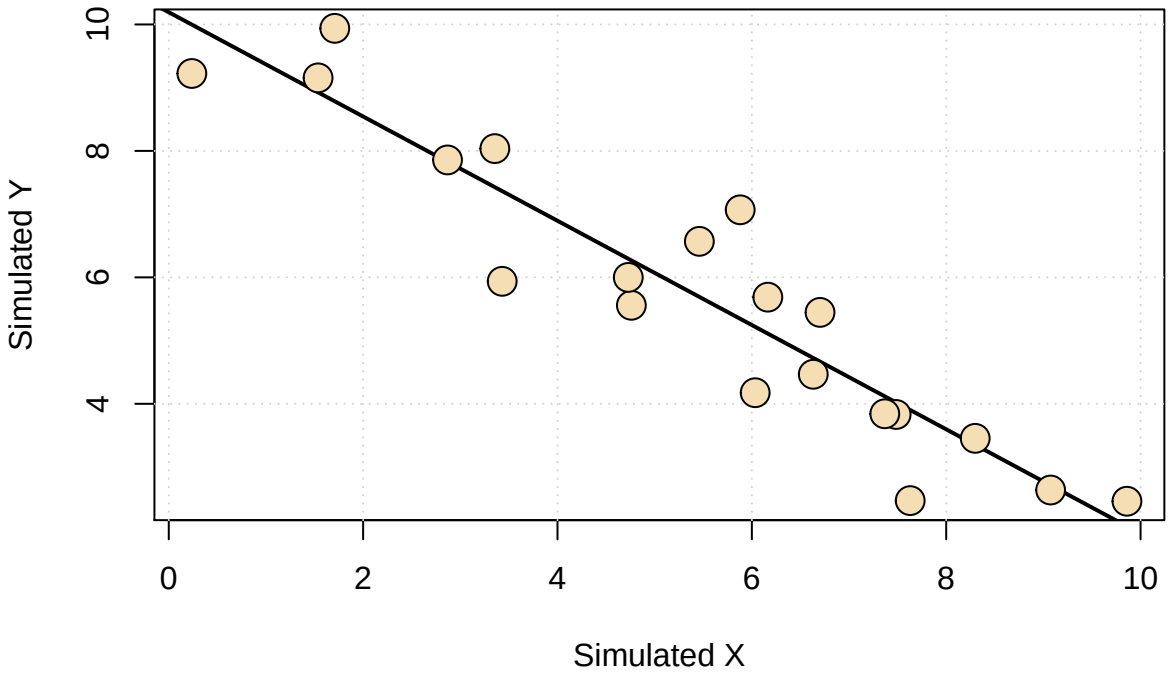
```
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```
z$coefficients[c(1,2)]
```

```
## [1] 10.310345 -0.862069
```

```
# set up model parameters
intercept <- 10.31 # fitted regression intercept
slope <- -0.86     # fitted regression slope
sampleSize <- 20           # simulated sample size
xRange <- c(0,10)  # permissable range of x-values
residSD <- 1       # standard deviation of residuals

RegSim <- function(a=intercept,b=slope,n=sampleSize,xl=xRange,sd=residSD,scatterplot=FALSE){

  # simulate x values from random uniform over range
  xSim <- runif(n=sampleSize,min=xRange[1],max=xRange[2])

  # simulate y values from regression plus sd for normal noise
  ySim <- a + b*xSim + rnorm(n=n,mean=0,sd=sd)

# create data frame of simulated data
SimData <- data.frame(cbind(xSim,ySim))
colnames(SimData) <- c("x","y")

# create and save summary of linear regression model
SimModel <- summary(lm(SimData$y~SimData$x))

# optionally, create regression plot
if(scatterplot==TRUE){

  plot(x=SimData$x,y=SimData$y,xlab= "Simulated X", ylab= "Simulated Y",type="n")
  grid()
  abline(SimModel,lwd=2)
  points(x=SimData$x,y=SimData$y,cex=2,pch=21,bg="wheat")
}

# create output as a list with data and model
output <- list(SimModel,SimData)

return(output)

}

RegSim(sd=1,scatterplot=TRUE)
```

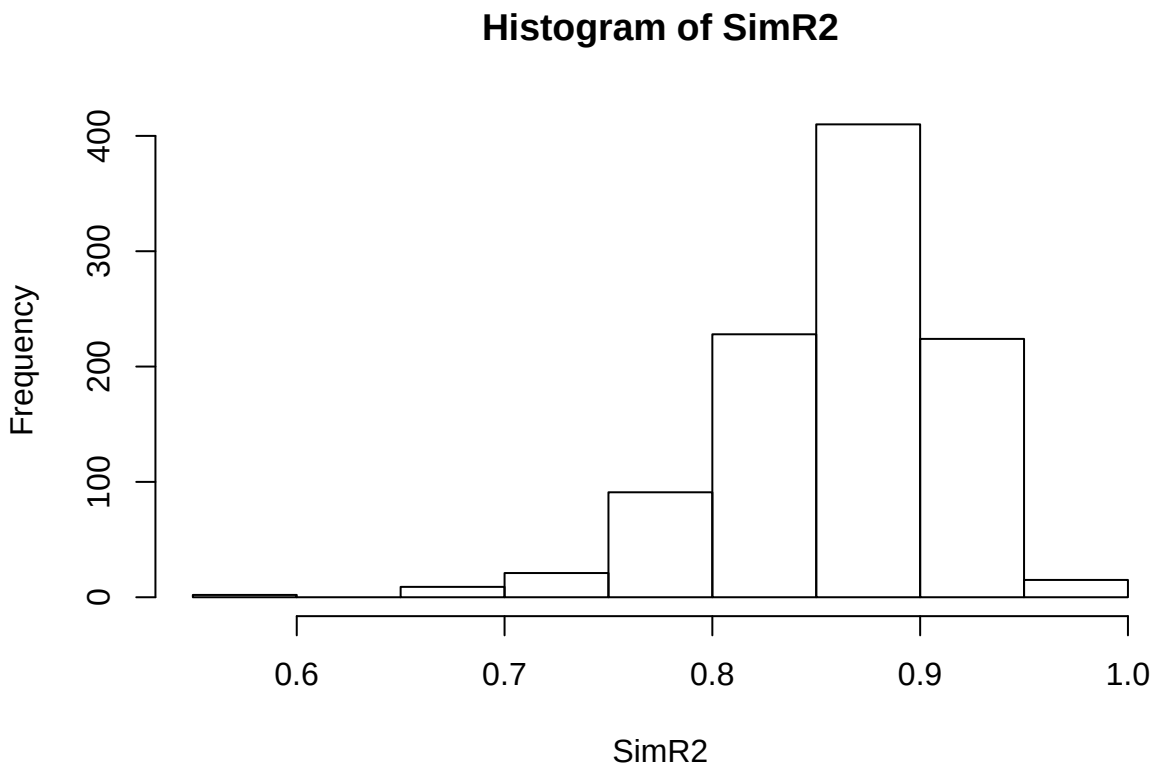
```
## Warning in abline(SimModel, lwd = 2): only using the first two of 8
## regression coefficients
```




```
## [[1]]
##
## Call:
## lm(formula = SimData$y ~ SimData$x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.43339 -0.39893 -0.02275  0.58481  1.72382
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.19311    0.45204   22.55 1.20e-14 ***
## SimData$x   -0.82456    0.07501  -10.99 2.04e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8549 on 18 degrees of freedom
## Multiple R-squared:  0.8703, Adjusted R-squared:  0.8631
## F-statistic: 120.8 on 1 and 18 DF,  p-value: 2.044e-09
##
##
## [[2]]
##           x        y
## 1  1.708133 9.938476
## 2  6.164366 5.686583
## 3  0.237257 9.224656
```

```
## 4  8.298412 3.453781
## 5  7.484622 3.831316
## 6  4.760070 5.558157
## 7  5.460008 6.569607
## 8  5.880177 7.068359
## 9  7.367821 3.842186
## 10 6.701637 5.445836
## 11 9.074515 2.636171
## 12 3.355260 8.036592
## 13 2.868279 7.856975
## 14 9.859949 2.458924
## 15 4.728406 5.998999
## 16 3.430857 5.937131
## 17 6.631865 4.466936
## 18 1.536420 9.155293
## 19 7.629394 2.468811
## 20 6.034341 4.175454
```

```r
# tweak the noise in the model to get r2 = 0.44

replicates <- 1000
SimR2 <- rep(0,replicates)

for (i in 1:replicates) {
  SimR2[i] <- RegSim(sd=1,scatterplot=FALSE)[[1]]$r.squared
}

hist(SimR2)
```




```
mean(SimR2)
```

```
## [1] 0.8615872
```

```
quantile(SimR2,probs=c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.7424873 0.9448751
```